

VB下如何编写CRC校验程序 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/459/2021_2022_VB_E4_B8_8B_E5_A6_82_E4_BD_c97_459710.htm 随着计算机技术的不断发展

，在现代工业中，利用微机进行数据通讯的工业控制应用得也越来越广泛。由于传输距离、现场状况等诸多可能出现的因素影响，计算机与受控设备之间的通讯数据常会发生无法预测的错误。为了防止错误所带来的影响，一般在通讯时采取数据校验的办法，而循环冗余码校验是最常用的校验方法之一。

一、循环冗余码校验原理 循环冗余码校验英文名称为Cyclical Redundancy Check，简称CRC。它是利用除法及余数的原理来作错误侦测（Error Detecting）的。实际应用时，发送装置计算出CRC值并随数据一同发送给接收装置，接收装置对收到的数据重新计算CRC并与收到的CRC相比较，若两个CRC值不同，则说明数据通讯出现错误。根据应用环境与习惯的不同，CRC又可分为以下几种标准：

CRC-12码；
CRC-16码；
CRC-CCITT码；
CRC-32码。CRC-12码通常用来传送6-bit字符串。CRC-16及CRC-CCITT码则是来传送8-bit字符，其中CRC-16为美国采用，而CRC-CCITT为欧洲国家所采用。CRC-32码大都被采用在一种称

为Point-to-Point的同步传输中。下面以最常用的CRC-16为例来说明其生成过程。CRC-16码由两个字节构成，在开始时CRC寄存器的每一位都预置为1，然后把CRC寄存器与8-bit的数据进行异或，之后对CRC寄存器从高到低进行移位，在最高位（MSB）的位置补零，而最低位（LSB，移位后已经被移出CRC寄存器）如果为1，则把寄存器与预定义的多项式码

进行异或，否则如果LSB为零，则无需进行异或。重复上述的由高至低的移位8次，第一个8-bit数据处理完毕，用此时CRC寄存器的值与下一个8-bit数据异或并进行如前一个数据似的8次移位。所有的字符处理完成后CRC寄存器内的值即为最终的CRC值。下面为CRC的计算过程：1. 设置CRC寄存器，并给其赋值FFFF(hex)。2. 将数据的第一个8-bit字符与16位CRC寄存器的低8位进行异或，并把结果存入CRC寄存器。3. CRC寄存器向右移一位，MSB补零，移出并检查LSB。4. 如果LSB为0，重复第三步；若LSB为1，CRC寄存器与多项式码相异或。5. 重复第3与第4步直到8次移位全部完成。此时一个8-bit数据处理完毕。6. 重复第2至第5步直到所有数据全部处理完成。7. 最终CRC寄存器的内容即为CRC值。

二、循环冗余码校验程序的编写明白了CRC校验码的产生过程，编写起程序来就非常容易了。由于Visual Basic的广泛普及以及其在数据通讯中的重要地位，下面就以VB语言来编写CRC的生成程序，其它语言只需稍做修改即可。编写CRC校验程序有两种办法：一种为计算法，一种为查表法。下面对两种方法分别讨论。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com